REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
10/8/96

3. REPORT TYPE AND DATES COVERED
Final Report 5/14/94-5/14/96

4. TITLE AND SUBTITLE
Neural Network Computing Architectures of Coupled Associative Memories with Dynamic Attractors

5. FUNDING NUMBERS
F
~~AFOSR~~49620-94-1-0239

6. AUTHOR(S)
Morris W. Hirsch

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Center For Pure and Applied Mathematics
University of California
Berkeley, CA 94720

AFOSR-TR-96
0559

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research/NM
110 Duncan Avenue Suite B115
Bolling Air Force Base, Washington, D.C. 20332-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
F
~~AFOSR~~49620-94-1-0239

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

In this time period, previous work on the construction of an oscillating neural network "computer" that could recognize sequences of characters of a grammar was extended to employ selective "attentional" control of synchronization to direct the flow of communication and computation within the architecture. This selective control of synchronization was used to solve a more difficult grammatical inference problem than we had previously attempted.

Further performance improvement was demonstrated by the use of a temporal context hierarchy in the hidden and context units of the architecture. These form a temporal counting hierarchy which allows representations of the input variations to form at different temporal scales for learning sequences with with long temporal dependencies.

We further explored the analog system identification capabilities of these systems where the output modules take on analog values. We were able to learn a mapping from the acoustic cepstral values of speech to articulatory parameters such as jaw and lip movement. This is a model speech processing problem which allows us to test the usefulness of our systems for speech recognition preprocessing.

19961125 185

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT

18. SECURITY CLASSIFICATION OF THIS PAGE

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT

DTIC QUALITY INSPECTED 3

FINAL TECHNICAL REPORT

Neural Network Computing Architectures of Coupled Associative Memories with Dynamic Attractors

Morris W. Hirsch (PI)
Dept Mathematics, U.C.Berkeley,
Berkeley, Ca. 94720, 510-642-4318

Bill Baird (Assist. Research Mathematician)
Dept Mathematics, U.C. Berkeley

Mathematics resource manager: Bernice Gangale 510-642-0116

Period: 5/15/94 - 5/14/96

Principal Investigator - Morris Hirsch

Department chair - C. Moore

Director of the Center for Pure and Applied Math - Alexandre Chorin

Contact and Grant officer - Patricia Gates

Abstract

In this time period, previous work on the construction of an oscillating neural network "computer" that could recognize sequences of characters of a grammar was extended to employ selective control of synchronization to direct the flow of communication and computation within the architecture, and a temporal context hierarchy was implemented for improved learning performance. Two papers and one book chapter were published [20, 12, 5], and presentations were made at three conferences.

The selective control of synchronization was used to solve a more difficult grammatical inference problem. Synchronization control is modeled as a subset of the hidden modules with ouputs which affect the resonant frequencies of other hidden modules. They learn to perturb these frequencies to control synchrony among these modules and direct the flow of computation to effect transitions between subsections of a large automaton which the system learns to emulate. The internal crosstalk noise is used to drive the required random transitions of the automaton.

In this architecture, oscillation amplitude codes the information content or activity of a module (unit), whereas phase and frequency are used to "softwire" the network. Only synchronized modules communicate by exchanging amplitude information. The same hardware and connection matrix can thus subserve many different computations and patterns of interaction between modules.

Even though it is constructed from a system of continuous nonlinear ordinary differential equations, the system can operate as a discrete-time symbol processing architecture, but with analog input and oscillatory subsymbolic representations.

We further explored the analog system identification capabilities of these systems where the output modules take on analog values. We had suprising success at learning a mapping from the acoustic cepstral values of speech to articulatory parameters such as jaw and lip movement. This is a model speech processing problem which allows us to test the usefulness of our systems for speech recognition preprocessing.

We showed further performance improvement by the use of a temporal context hierarchy in the hidden and context units of our architecture. The hidden and context units of our cortical architecture are grouped so that there is a hierarchy of sets which only change attractors at increasing multiples of the base clock cycle. These then form a temporal counting hierarchy which allows representations of the input variations to form at different temporal scales for learning sequences with with long temporal dependencies.

Because intercommunicating modules of the architecture are analytically guaranteed to store and recall multiple oscillatory and chaotic attractors, the architecture serves as a framework in which to arrange and exploit the special capabilities dynamic attractors.

Chaotic attractors from the large family of Chua attractors were synchronized for operation in the architecture using techniques of coupling developed for secure "broadspectrum" communication by a modulated chaotic carrier wave.

This type of computing architecture and its learning algorithms for computation with oscillatory spatial modes is ideal for implementation in optical systems, where electromagnetic oscillations, very high dimensional modes, and high processing speeds are available. The mathematical expressions for optical mode competition are identical to our normal form equations for oscillatory mode competition.

1 Introduction

We have shown analytically and numerically how a neural network "computer" architecture, inspired by the structure of cerebral cortex, can be constructed of recurrently interconnected associative memory modules of the type developed in last years work. The architecture is such that the larger system is itself a special case of the type of network of the modules, and can be analysed with the same tools used to design the subnetwork modules.

The modules in the architecture can learn connection weights between themselves which cause the system to evolve under a clocked "machine cycle" by a sequence of transitions of attractors within the modules, much as a digital computer evolves by transitions of its binary flip-flop states. Thus the architecture employs the principle of "computing with attractors" used by macroscopic systems for reliable computation in the presence of noise. Clocking is done by rhythmic variation of certain bifurcation parameters which hold some modules clamped at their attractors while others transition.

We have constructed a discrete-time recurrent "Elman" network architecture with oscillatory modules. The time steps (machine cycles) of the system hold input and "context" modules clamped at their oscillatory attractors while "hidden" modules change state, then clamp hidden states while context modules are released to load those states as the new context for the next cycle of input.

The capabilities of this architecture were explored by application to the well studied problem of grammatical inference. We have at present a system which functions as a finite state automaton that perfectly recognizes or generates the infinite set of six symbol strings that are defined by a Reber grammar. Even though it is constructed from a system of continuous nonlinear ordinary differential equations, the system can operate as a discrete-time symbol processing architecture, but with analog input and oscillatory subsymbolic representations.

Most recently we have shown how the Elman architecture can learn to employ control of selective synchronization to direct the flow of communication and computation within the architecture to solve a grammatical inference problem.

In this architecture, oscillation amplitude codes the information content or activity of a module (unit), whereas phase and frequency are used to "softwire" the network. We have shown that only synchronized modules communicate by exchanging amplitude information; the activity of non-resonating modules is shown to contribute noise. The same hardware and connection matrix can thus subserve many different computations and patterns of interaction between modules.

Synchronization control is modeled as a subset of the hidden modules with ouputs which affect the resonant frequencies of other hidden modules. They learn to perturb these frequencies to control synchrony among these modules and direct the flow of computation to effect transitions between subsections of a large automaton which the system learns to emulate. The internal crosstalk noise is used to drive the required random transitions of the automaton.

Because intercommunicating modules of the architecture can store and recall multiple oscillatory and chaotic attractors, the architecture can serve as a framework in which to arrange and exploit the special capabilities dynamic attractors. We have therefore also synchronized chaotic attractors from the large family of Chua attractors for operation in the architecture using techniques of coupling developed for secure "broadspectrum" communication by a modulated chaotic carrier wave.

Since our modules can operate in an analog mode, we have begun to explore the map learning capabilities of these systems where the output modules take on analog values. We have had suprising early success at learning a mapping from the acoustic cepstral values of speech to articulatory parameters such as jaw and lip movement. This is a model speech processing problem which allows us to test the usefulness of our systems for speech recognition preprocessing.

We also investigated the use of a *temporal context hierarchy* in learning sequences like this with with long temporal dependencies. The hidden and context units of our cortical architecture are grouped so that there is a hierarchy of sets which only change attractors at increasing multiples of the base clock cycle. These then form a temporal counting hierarchy which allows representations of the input variations to form at different temporal scales.

In summary, the architecture is designed to demonstrate and study the following issues and principles of neural computation:

- Sequential computation with coupled associative memories.

- Computation with attractors for reliable operation in the presence of noise.
- Combined advantages of attractor associative memory networks and recurrent multilayer connectionist networks.
- Operation of associative memories near self organized multiple critical points for bifurcation control of attractor transitions.
- Discrete time and state symbol processing arising from continuum dynamics by bifurcations of attractors.
- Hybrid analog and symbolic computation.
- Temporal context hierarchy for learning of extended temporal dependencies.
- Attention as selective synchronization of dynamic attractors controling communication and temporal program flow.
- Broadspectrum synchronization of chaotic attractors
- Chaotic search - chaotic dynamics driving random choice of attractors in network modules.

To advance intuition for theoretical analysis, interactive simulations of the network applications have been designed on the SGI 4D35G Personal Iris Graphics Workstation. These allow real time graphic display of network dynamics and learning as parameters are varied.

2 Normal Form Associative Memory Modules

The mathematical foundation for the construction of network modules of this architecture is contained in the **normal form projection algorithm**[2, 9].

The normal form projection algorithm , developed at U.C.Berkeley, allows analytically guaranteed associative memory storage of analog patterns, continuous sequences, and chaotic attractors in the same network. An N node network can be shown to store up to N static, $N/2$ oscillatory, or $N/3$ chaotic memory attractors [2, 9, 6]. Single modules with either static, oscillatory, or one of four types of chaotic attractors – Lorenz, Roessler, Ruelle-Takens,Chua – have been sucessfully used for recognition of handwritten characters [4, 10, 20].

A key feature of a net constructed by this algorithm is that the underlying dynamics is explicitly isomorphic to any of a class of standard, well understood nonlinear dynamical systems - a "normal form" [19]. This system is chosen in advance, independent of both the patterns to be stored and the learning algorithm to be used. This control over the dynamics permits the design of important aspects of the network dynamics independent of the particular patterns to be stored. Stability, basin geometry, and rates of convergence to attractors can be investigated and programmed in the standard dynamical system.

The network modules of this architecture were developed previously as models of olfactory cortex [1, 3]. In this biological model, the attractors within modules are distributed patterns of activity like those observed experimentally [18]. However, the network is equivalent to the architecture of modules in normal form and may easily be designed, simulated, and theoretically evaluated in these coordinates.

By analyzing the network in the polar form of these "normal form coordinates", the amplitude and phase dynamics have a particularly simple interaction. When the input to a module is synchronized with its intrinsic oscillation, the amplitudes of the periodic activity may be considered separately from the phase rotation, and the network of the module may be viewed as a static network with these amplitudes as its activity. We have further shown analytically that the network modules we have constructed have a strong tendency to synchronize as required.

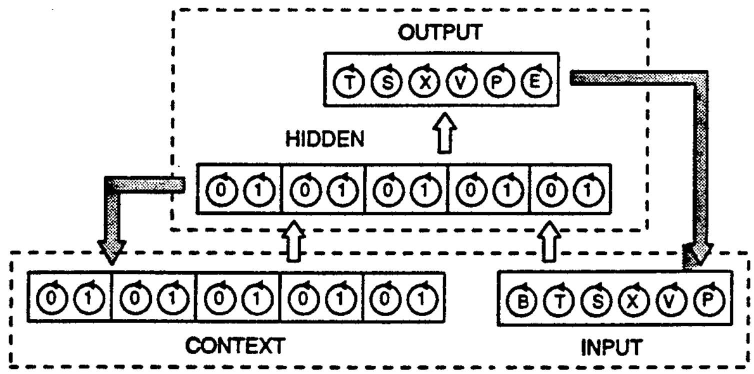

Figure 1: Elman architecture: The input and output layer each consist of a single associative memory module with six oscillatory attractors, one for each of the six symbols in the grammar. The hidden and context layers consist of binary "units" composed of two oscillatory attractors.

3 Elman Architecture

As a benchmark for the capabilities of the system, and to create a point of contact to standard network architectures, we have shown how a discrete-time recurrent "Elman" network architecture [16] can be constructed from recurrently connected oscillatory associative memory modules described by continuous nonlinear ordinary differential equations [11, 8].

The time steps (machine cycles) of the system are implemented by rhythmic variation (clocking) of a bifurcation parameter. This holds input and "context" modules clamped at their attractors while 'hidden and output modules change state, then clamps hidden and output states while context modules are released to load those states as the new context for the next cycle of input.

We use two types of modules in implementing the Elman network architecture. The input and output layer each consist of a single associative memory module with six oscillatory attractors (six competing oscillatory modes), one for each of the six possible symbols in the grammar. The hidden and context layers consist of binary "units" composed of a two oscillator module with internal competition. We think of one mode within the unit as representing "1" and the other as representing"0" (see figure 2).

The network approximates a static network unit in its amplitude activity when fully phase-locked. Amplitude information is transmitted between modules, with an oscillatory carrier. If the frequencies of attractors in the architecture are randomly dispersed by a significant amount phase-lags appear, then synchronization is lost and improper transitions begin to occur.

The ability to operate as an finite automaton with oscillatory/chaotic "states" is thus an important benchmark for this architecture, but only a subset of its capabilities. At low to zero competition, the supra-system reverts to one large continuous dynamical system. We expect that this kind of variation of the operational regime, especially with chaotic attractors inside the modules, though unreliable for habitual behaviors, may nontheless be very useful in other areas such as the search process of reinforcement learning.

4 Synchronization, Noise, and Intermodule Communication

An important element of intra-cortical communication in the brain, and between modules in this architecture, is the ability of a module to detect and respond to the proper input signal from a particular module, when inputs from other modules which is irrelevant to the present computation are contributing cross-talk and noise. This is smilar to the problem of coding messages in a computer architecture like the Connection Machine so that they can be picked up from the common communication buss line by the proper receiving module. We are investigating the hypothesis that sychronization control is one way the brain can solve this coding problem.

Because communication between modules in the architecture is by continuous time-varying analog vectors, the process is more one of signal detection and pattern recognition by the modules of their inputs than it is "message passing". This is why the demonstrated performance of the modules in handwritten character

recognition is significant, and why we expect there are important possibilities in the architecture for the kinds of chaotic signal processing studied by Chua [17].

We have shown that the dynamic attractors - oscillatory or chaotic - within the modules of this architecture must synchronize to effectively communicate information and produce reliable transitions [7]. In these simulations, we synchronized lorenz and Chua attractors for operation in the architecture using techniques of coupling developed by Chua [17] for secure "broadspectrum" communication by a modulated chaotic carrier wave [8, 10].

5 Control of Synchrony

The network architecture, shown in figure 5, has been designed so that amplitude codes the information content or activity of a module, whereas phase and frequency are used to "softwire" the network. An oscillatory network module has a passband outside of which it will not synchronize with an oscillatory input. Modules can therefore easily be desynchronized by perturbing their resonant frequencies. Furthermore, only synchronized modules communicate by exchanging amplitude information; the activity of non-resonating modules contributes incoherant crosstalk or noise. The flow of communication between modules can thus be controled by controlling synchrony. By changing the intrinsic frequency of modules in a patterned way, the *effective* connectivity of the network is changed. The same hardware and connection matrix can thus subserve many different computations and patterns of interaction between modules without crosstalk problems.

The crosstalk noise is actually essential to the function of the system. It serves as the noise source for making random choices of output symbols and automaton state transitions in this architecture during reinforcement learning and normal operation after learning. In cortex there is an issue as to what may constitute a source of randomness of sufficient magnitude to perturb the behavior of the large ensemble of neurons involved in neural activity at the cortical network level. It does not seem likely that the well known molecular level of fluctuations which is easily averaged within a single neuron or small group of neurons can do the job. The architecture here models the hypothesis that deterministic chaos in the macroscopic dynamics of a network of neurons, which is the same order of magnitude as the coherant activity, can serve this purpose.

In a set of modules which is desynchronized by perturbing the resonant frequencies of the group, coherance is lost and "random" phase relations result. The character of the model time traces is now irregular as seen in real neural ensemble activity. The behavior of the time traces in different modules of the architecture is similar to the temporary appearance and switching of synchronization between cortical areas as seen in observations of cortical processing during sensory/motor tasks in monkeys and humans [14]. The detailed structure of this apparently chaotic signal and its further use in network learning and operation are currently under investigation.

6 Grammatical Inference

We studied the use of these capabilities in the grammatical inference problem by constructing and learning the larger fifteen hidden unit (module) automata studied by Cleermans, et al, shown in figure 5. This consists of two subgraphs each of which was the automaton learned previously in work described above. Strings of this grammar can contain long embedded sequences of the smaller grammar before the final transition distinguishing which branch you are on appears. These transitions of this grammar were challenging to learn because of the embedding. Cleermans *et al* had to alter the transition probabilities within the two smaller automata so that the backpropagation algorithm could distinguish the branches during learning.

We solved this learning problem by introducing a control of program flow by selective synchronization [13]. The controler itself is modeled in this architecture as a special set of hidden modules with ouputs that affect the resonant frequencies of the other hidden modules.

These enforce a segregation of the hidden module code for the subautomata states during training so that different sets of synchronized modules learn to code for each subautomata with the other modules desynchronized by frequency perturbation. The entire automaton is learned with its additional entry and exit hidden module states and with these special hidden modules.

The system in operation can be made to jump from states in one subautomaton to the other by desynchronizing the proper subset of hidden modules. The possibilities for transition of the system can thus

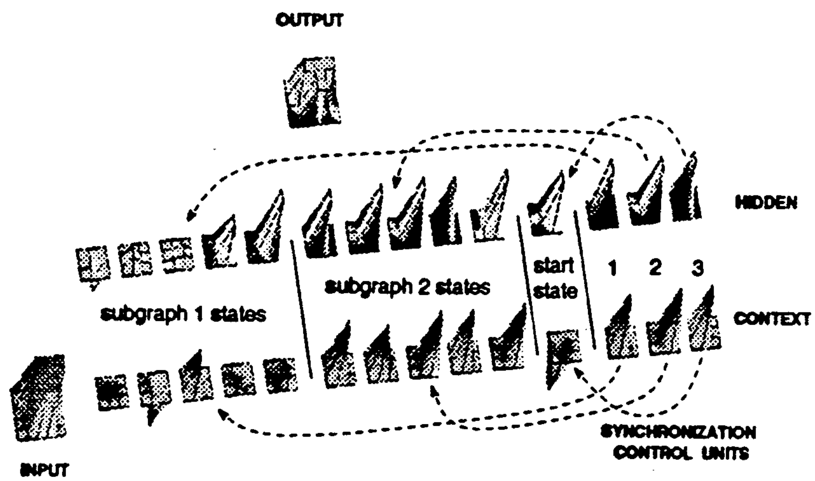

Figure 2: Synchronization control architecture: The input and output modules show the symbol "T" as a distributed attractor pattern. The binary modules of the hidden and context layers show oscillatory attractors in winner-take-all normal form cordinates where one oscillator at its maximum amplitude, with the others near zero amplitude. Activity levels oscillate up and down through the plane of the paper. Dotted lines show control outputs from the synchronization control modules. Control unit two is at the one attractor (right side of the square active) and the hidden units coding for states of subgraph two are in synchrony with the input and output modules. Here in midcycle, all modules are clamped at their attractors.

be controled by selective synchronization. This control itself is learned by the special hidden units whose output controls the synchrony of these subsets. During training, the control modules learn to respond to the proper input symbol and context to direct the flow of computation to effect the difficult transitions between subautomata. Viewing the automata above as a behavioral program, the control of synchrony constitutes a control of the program flow into its subprograms (the subautomata).

7 Acoustic-to-Articulatory Parameter Estimation

We have selected a model speech processing problem which allows us to test the usefulness of our systems for speech recognition preprocessing. Since our modules can operate in analog mode, we are exploring the map learning capabilities of these systems where the output modules take on analog values. We are attempting to learn a mapping from the acoustic cepstral values of speech to articulatory parameters such as jaw and lip movement. This is less difficult than the segmentation and recognition of phonemes required for speech recognition, but the problems of forward and backward coarticulation are still encountered. The system can be used in speech recognition preprocessing to solve the "cocktail party" problem and for speaker independent speech recognition as discussed below. The task will test the the capability of the self-organizing context of our system to disambiguate the one-to-many acoustic-to-articulation map based on its sensitivity to past history and predicition of the future.

As yet no one has used these discrete time recurrent nets for this parameter estimation task. Some groups have used nets to learn acoustic to articulatory parameters. Rahim, Goodyear, et. al. found a simple feedfoward net to be inadequate to handle coarticulation effects []. Shirai and Kobayashi used a feed forward 4 layer net to learn the mapping to lip, jaw, and tongue parameters for vowels.

Kobyashi learned articulatory movements, but only for vowels, and the Goodyear group predicted vocal tract areas, but couldn't handle coarticulation proplems in large data sets with a single net. These were both feedforward nets, and we expect that the internal feedback of our recurrent nets will give us the past and future context sensitivity needed for succeeding with a single net. A goal is to learn a real time mapping into multiple articulatory parameters for continuous single or speaker independent speech. With a combination of networks, the Goodyear group seems to have done this for single speaker sentences, but the system is not

real-time. Their super-network is large and requires dynamic programming to select optimal subnetwork results.

A major task is to get data for training. Kobyashi used nonlinear regression of the articulatory model (of Mermelstein) on labeled acoustic data to get synchronized acoustic and articulatory data streams. At present we have constructed a mechanical device for lip and jaw data. This is a spring loaded potentiometer that spreads two flat metal strips between the lips to follow the opening motion. This has given us data to start with for network testing purposes. We use a field effect transistor to modulate the amplitude of a digital timer oscillation at 5 k Hz to input the lip movement signal to the left channel of the stereo audio port in synchrony with the speech acoustic signal going into the right channel. Then we average the 128 samples taken per time window to recover the lip movement amplitude signal.

We use a time domain algorithm on the digitized speech signal to get 12 average linear predictive (LPC) coeficients over 128 samples in a 16 msec time window. Rabiner and Schaefer argue for the clean formant characterizing properties of these over other cepstral values. We use both slow vowel data and sentences of fast speech for our training data.

During training and we can watch the convergence of the learned network output onto the moving lip target where output and target are displayed as flat ellipses of different colors. Using a small database of 30 training sentences and vowels, and a crossvalidation test set of 20 sentences and vowels, we are getting striking results in networks of 3, 5, and 10 hidden units. Anyone can speak into the computer microphone and watch the lip model open and close in real time. With this one dimension of output, the system appears to follow fast speech of different speakers quite well, while trained on only one speaker. It does not always follow slow vowels and nonsense gestures accurately. We are producing a data set of vertical and horizontal lip opening to further test the system, and are waiting to obtain high dimensional lip model data from the lipreading project at the International Computer Science Institute in Berkeley.

Estimation of articulatory parameters has been used to normalize pitch and other differences between speakers to aid in speaker independent speech recognition []. Our efforts here may thus be of benefit in the recognition domain as well. Should our rhythm entrained recurrent networks prove sucessful on the voice tracking project, we hope to work with Nelson Morgan at the ICSI speech recognition effort to apply these systems to recognition problems.

8 Time Scale Hierarchy

We are investigating the use of a *temporal context hierarchy* in learning sequences with long temporal dependencies. The hidden and context units of our cortical architecture will be grouped to cross-inhibit each other so that there is a hierarchy of sets which only change attractors at increasing multiples of the base clock cycle. These then form a temporal counting hierarchy which allows representations of the input variations to form at different temporal scales.

Other work in simple recurrent networks using backpropagation [21] has shown that these slower-changing units learn to code for features of longer stretches of an input sequence. They then act as high level "hypotheses" which aid in the recognition of long sequences by retaining information about the earlier segments. During sequence *generation*, such high level modules may be viewed as "plans" that activate long sequences of lower level hidden unit transitions and output behaviors. Mozer has used hidden units with a hierarchy of decay times to learn the "grammar" of Bach organ solos, for generation of novel convincing Bach style passages [21].

In addition to the applications to the tasks described above, we will investigate this approach by learning the automaton (fifteen hidden units) studied by Cleermans *et al* [15] using this temporal hierarchy in the hidden units. We can then compare the results with our previous work [11, 13] to determine the performance benefits.

9 Computing Resources

Our analytic approach to understanding these networks relies heavily on geometric visualization of network learning and operation in prefered coordinate systems. The computer graphic capabilities of the Silicon Graphics Personal Iris 4D35G workstation purchased by the grant has been invaluable in enabling us to

design interactive simulations with graphical display of these geometric representations in order to enhance our intuition and generate new theoretical insights.

We have employed the workstation as a system for simulation and graphic display of network dynamics, where we can vary network parameters (most notably bifurcation parameters) and alter network dynamics in real time. With this capability, we were able to rapidly explore regions of the parameter space, and find where to concentrate our numerical and analytical efforts.

10 Invited Talks and Conferences

Society for Music Perception and Cognition, June '95
Computation and Neural Systems *95, San Francisco, Ca. July '95
Cognitive Neuroscience Meeting, San Francisco, Ca, March 31-April 2, 1996.

References

[1] B. Baird. Bifurcation and learning in network models of oscillating cortex. In S. Forest, editor, *Emergent Computation*, pages 365–384. North Holland, 1990. also in Physica D, 42.

[2] B. Baird. A learning rule for CAM storage of continuous periodic sequences. In *Proc. Int. Joint Conf. on Neural Networks, San Diego*, pages 3: 493–498, June 1990.

[3] B. Baird. Learning with synaptic nonlinearities in a coupled oscillator model of olfactory cortex. In F. Eeckman, editor, *Analysis and Modeling of Neural Systems*, pages 319–327, 1992.

[4] B. Baird, W. Freeman, F. Eeckman, and Y. Yao. Applications of chaotic neurodynamics in pattern recognition. In *SPIE Proceedings Vol. 1469*, pages 12–23, 1991.

[5] B. Baird, T. Troyer, and F. Eeckman. Attentional network streams of synchronized 40 hz activity in a cortical architecture of coupled oscillating associative memories. In D. Levine, V. Brown, and T. Shirley, editors, *Oscillations in Neural Systems.* Laurence Erlbaum, 1996. in press.

[6] Bill. Baird and Frank H. Eeckman. CAM storage of analog patterns and continuous sequences with $3n^2$ weights. In D. S. Touretsky, editor, *Advances in Neural Information Processing Systems 3*, pages 192–198. Morgan Kaufman, 1991.

[7] Bill Baird and Frank H. Eeckman. A hierarchical sensory-motor architecture of oscillating cortical area subnetworks. In Frank H. Eeckman, editor, *Analysis and Modeling of Neural Systems II*, pages 96–104, Norwell, Ma, 1992. Kluwer.

[8] Bill Baird and Frank H. Eeckman. A neural network computer architecture for computation with oscillatory and chaotic attractors. In Takeshi Yamakawa, editor, *Proc. 2nd Int. Conf. on Fuzzy Logic and Neural Networks, Iizuka, Japan*, pages 981–985, 1992.

[9] Bill Baird and Frank H. Eeckman. A normal form projection algorithm for associative memory. In Mohamad H. Hassoun, editor, *Associative Neural Memories: Theory and Implementation*, New York, NY, 1993. Oxford University Press.

[10] Bill Baird, Morris W. Hirsch, and Frank Eeckman. A neural network associative memory for handwritten character recognition using multiple chua attractors. *IEEE Trans. Circuits and Systems*, pages 667–675, 1993.

[11] Bill Baird, Todd Troyer, and Frank H. Eeckman. Synchronization and gramatical inference in an oscillating elman network. In S.J. Hanson, J.D. Cowan, and C.L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 236–244. Morgan Kaufman, 1993.

[12] Bill Baird, Todd Troyer, and Frank H. Eeckman. Attention as selective synchronization of oscillating cortical sensory and motor associative memories. In Frank H. Eeckman, editor, *Neural Systems Analysis and Modeling*, pages 167–175, Norwell, Ma, 1994. Kluwer.

[13] Bill Baird, Todd Troyer, and Frank H. Eeckman. Gramatical inference by attentional control of synchronization in an oscillating elman network. In S.J. Hanson, J.D. Cowan, and C.L. Giles, editors, *Advances in Neural Information Processing Systems 6*, pages 67–75. Morgan Kaufman, 1994.

[14] S. Bressler and Nakamura. Inter-area synchronization in macaque neocortex during a visual pattern discrimmination task. In F.H. Eeckman and J. Bower, editors, *Neural Systems: Analysis and Modelling*, page 515, Norwell, MA, 1993. Kluwer Academic Publishers.

[15] Axel Cleeremans, David Servan-Schreiber, and James McClelland. Finite state automata and simple recurrent networks. *Neural Computation*, 1(3):372–381, 1989.

[16] J.L. Elman. Distributed representations, simple recurrent networks and grammatical structure. *Machine Learning*, 7(2/3):91, 1991.

[17] Tetsuro Endo and Leon O. Chua. Synchronization and chaos in phase-locked loops. *IEEE Transactions on Circuits and Systems*, 38(12):620–626, 1991.

[18] W.J. Freeman and B. Baird. Relation of olfactory EEG to behavior: Spatial analysis. *Behavioral Neuroscience*, 101:393–408, 1987.

[19] J. Guckenheimer and D. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields.* Springer, New York, 1983.

[20] M. W. Hirsch and B. Baird. Computing with dynamic attractors in neural networks. *Biosystems*, 1993. to appear.

[21] Michael C. Mozer. Induction of multiscale temporal structure. In John E. Moody, Steven J. Hanson, and Richard P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 275–282. Morgan Kaufman, 1992.